

AD-A081 968

SRI INTERNATIONAL MENLO PARK CA  
FINANCIAL MANAGEMENT SYSTEM. VOLUME I. (U)  
JAN 80 R J ROM, H G LEHTMAN

F/6 9/2

F30602-78-C-0055

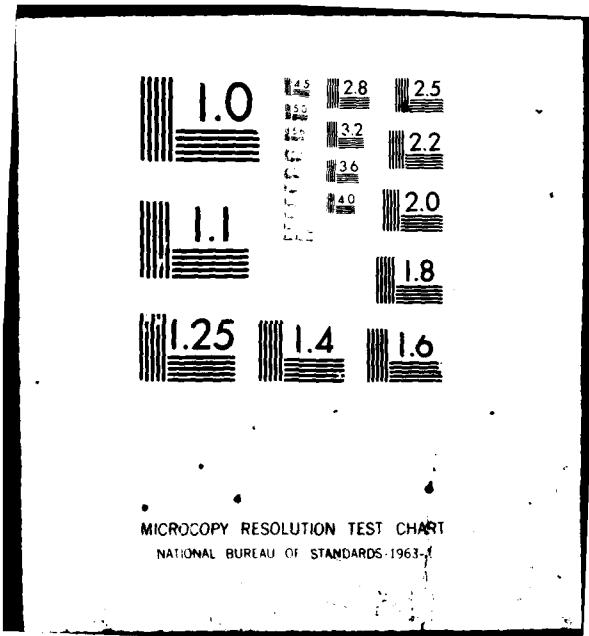
UNCLASSIFIED

RADC-TR-79-299-VOL-1

ML

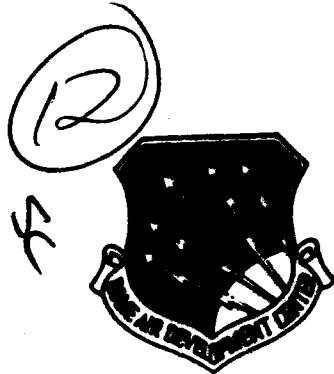
for  
all  
use  
and  
copy  
right  
reserved

END  
DATE  
FILED  
4-80  
OTIC



1  
ADA 081968

LEVEL A



RADC-TR-79-299, Vol 1 (of two)  
Final Technical Report  
January 1980

## FINANCIAL MANAGEMENT SYSTEM

SRI International

Raphael J. Rom  
Harvey G. Lehtman

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DTIC  
ELECTED  
MAR 18 1980  
S D  
A

ROME AIR DEVELOPMENT CENTER  
Air Force Systems Command  
Griffiss Air Force Base, New York 13441

DC FILE COPY

80 3 -17 226

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-79-299, Vol I (of two) has been reviewed and is approved for publication.

APPROVED:

RICHARD CALICCHIA  
Project Engineer

APPROVED:

WENDALL C. BAUMAN, Col, USAF  
Chief, Information Sciences Division

FOR THE COMMANDER:

JOHN P. HUSS  
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (ISIE), Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

## UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ⑯ RADC-TR-79-299, Vol I (of two)	2. GOVT ACCESSION NO. ⑯	3. RECIPIENT'S CATALOG NUMBER ⑯
4. TITLE (and Subtitle) ⑯ FINANCIAL MANAGEMENT SYSTEM • Volume I.	5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 18 Jan 78 - 9 Aug 79	
7. AUTHOR(s) ⑯ Raphael J. Rom Harvey G. Lehtman	8. PERFORMING ORG. REPORT NUMBER N/A	
9. PERFORMING ORGANIZATION NAME AND ADDRESS SRI International / 333 Ravenswood Avenue Menlo Park CA 94025	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ⑯ 62702F 55812006 ⑯ 17) 20	
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (ISIE) Griffiss AFB NY 13441	12. REPORT DATE Jan 1980	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same ⑯ TR-79-299-VOL-1	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same		
18. SUPPLEMENTARY NOTES RADC Project Engineer: Richard Calicchia (ISIE)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) On-line FMS software engineering augment programming environments computers system software		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This contract was initiated to remedy inefficiency on the prototypical Financial Management System (FMS) under a previous development and to implement techniques for the handling of copious data. This was accomplished by converting the system from running under NLS version 8.5 to Augment NLS version 10 (NLS 10). Important capabilities were added to the system such as sophisticated search, simulation and ledger tracing. The efficiency was increased by improving the algorithms in order to provide better safeguard for data integrity. A mechanism was developed to generate data from several sources which provided (Cont'd)		

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

410281 \*21

**UNCLASSIFIED**

**SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)**

**Item 20 (Cont'd)**

a sophisticated report generating facility including the manpower data base. A comprehensive documentation was also provided for both casual users and professional programmers.

Accession For	
NTIS GRANT	
L72 TAB	
UNCLASSIFIED	
Distribution	
By	
Date	
Filing Date	
Filing Location	
Filing Type	
DRAFT Special	

UNCLASSIFIED

**SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)**

## TABLE OF CONTENTS

INTRODUCTION . . . . .	1
FINAL TECHNICAL REPORT SUMMARY . . . . .	2
PURPOSE OF PROJECT . . . . .	2
IMPORTANT SYSTEMS DESIGNED, DEVELOPED, AND DOCUMENTED . . . . .	3
Introduction . . . . .	3
FMS/DES Subsystems . . . . .	3
Manpower Subsystem . . . . .	3
Template/Forms System . . . . .	3
Modifications to AUGMENT . . . . .	4
Documentation . . . . .	4
CONCLUSIONS REACHED . . . . .	5
PROBLEMS ENCOUNTERED . . . . .	6
GENERAL METHODOLOGY . . . . .	7
TECHNICAL RESULTS . . . . .	7
IMPLICATIONS FOR FURTHER RESEARCH . . . . .	8
EVALUATION OF PROBLEM AREAS . . . . .	9
FMS/DES SUBSYSTEMS . . . . .	9
Restructuring FMS . . . . .	10
Improved User Capabilities . . . . .	14
MANPOWER SUBSYSTEM . . . . .	17
TEMPLATE/FORMS SYSTEM . . . . .	18
MODIFICATIONS TO AUGMENT . . . . .	20
DOCUMENTATION . . . . .	21

APPENDIX I SUMMARY OF NEW COMMANDS AND FEATURES IN FMS/DES . . . . .	22
APPENDIX II THE MANPOWER SUBSYSTEM FORM2 FORMAT AND COMMANDS . . . . .	23
APPENDIX III TEMPLATE LANGUAGE SYNTAX AND SAMPLE RADC 74 TEMPLATE . . . . .	28

## EVALUATION

This effort was performed to improve the maintenance and flexibility of the existing version of on-line information data base for Financial Management System (FMS). This was achieved by implementing various extensions to document the system more completely and to assist in integrating FMS into the RADC working environment. The present effort was the final phase to eliminate certain inefficiencies existing on earlier developments under previous contracts.

The work was accomplished by converting the system from the previous NLS version 8.5 to the newly developed Augment version NLS 10. This improved the algorithm for increased efficiency and provided better safeguard for data integrity for the Data Entry System (DES).

The results were successful. It added important capabilities to the system such as sophisticated search, simulation and ledger tracing. It provided a sophisticated report generating facility. With this a mechanism was developed to generate data from several sources, especially the manpower data base. Lastly, it provided a comprehensive documentation for both casual users and professional programmers.



RICHARD CALICCHIA  
Project Engineer

## INTRODUCTION

This report is submitted in fulfillment of the requirement for a Final Technical Report as listed in the Contract Data Requirements List Item A006 of the Financial Management System project contract. (The title of the Statement of Work for this project was "NLS Applications Programming Development". The work on this project was covered by RADC-SRI Contract Number F30602-78-C-0055 and SRI-Tymshare Subcontract Number 14392.)

Final Technical Report Summary:  
Purpose of Project

FINAL TECHNICAL REPORT SUMMARY

PURPOSE OF PROJECT

SRI International and Tymshare's Augmentation Resources Center (ARC) had developed a prototypical Financial Management System (FMS) under an earlier contract. This system was an online information system for maintaining, accessing, and analyzing a financial data base. Although the first FMS served its purpose as a prototype, there were certain inefficiencies in its handling of data which had to be solved, as well as several desirable extensions which could be added. The current effort resulted in a system that attempted to remedy most of these inefficiencies and implement some of the possible extensions. Specifically, the more important goals were to:

Convert the system from running under NLS version 8.5 to the newly developed AUGMENT system. (AUGMENT, otherwise known as NLS10, is the latest version of the NLS system developed at SRI International and, more recently since ARC's acquisition, at Tymshare.)

Add important capabilities to the system such as sophisticated search, simulation, and ledger tracing.

Improve the algorithms to increase efficiency and provide better safeguards for data integrity.

Provide a sophisticated report generating facility.

Develop a Manpower system for collecting information on time spent on various projects which could be incorporated into the general FMS data base.

Develop a mechanism to generate data from several sources, including the Manpower data base.

Provide comprehensive documentation for both casual users and professional programmers.

Final Technical Report Summary:  
Important Systems Designed, Developed, and Documented

**IMPORTANT SYSTEMS DESIGNED, DEVELOPED, AND DOCUMENTED**

**Introduction**

Design, implementation, and documentation work has been undertaken on the following tasks as part of the FMS project. They all fit into the context of the AUGMENT system environment.

**FMS/DES Subsystems**

FMS exists as two integrated subsystems under AUGMENT. The first subsystem, the Data Entry System (DES), is a data base tool that maintains precise control over the form and content of every data entry in a structured data base. DES use is limited to qualified data base users and is unavailable to the general user; it is the only subsystem that can write information to the data base. The second subsystem is a user's tool, referred to as "FMS" because it is the most commonly used FMS subsystem. FMS allows direct user interaction, including retrieval and report generation, with the data maintained by FMS. It can only read and not change information in the data base.

Specific extensions to FMS/DES carried out under this contract are outlined in Appendix I.

**Manpower Subsystem**

The Manpower subsystem is an integral part of FMS that records mainly manpower-related data. It is distinguished from the general FMS/DES subsystems by having a separate, more widely used, data entry mechanism. Portions of the Manpower data eventually become an integral part of FMS and are available to FMS users in all retrieval operations.

The Manpower data entry subsystem, referred to as "FORM2", is designed to record, modify, transfer, and view employee work time in specified Efforts during a month's period and, after the month's period, to integrate summaries of this data into the FMS data base.

**Template/Forms System**

ARC has continued work on and modified a general purpose form generation system. Part of this system is a template writing language for specifying locations of information on pages or terminals and styles of data collection from users or data management systems. It also permits the establishment of contingencies among fields in one or more forms. The Template system is designed so

Final Technical Report Summary:  
Important Systems Designed, Developed, and Documented

that users will ultimately be able to write their own templates and modify existing ones. The AUGMENT version of the FMS system makes use of this system to generate reports.

A modified sample RADC 74 (JB form) template was created to show the feasibility of generating reports automatically from the FMS data base. Special features of the Template system were used to create data elements that obtained information from the FMS data base.

#### Modifications to AUGMENT

In order to implement the systems developed under this contract, some minor modifications had to be made to AUGMENT. One of these was the modification of address spaces and procedures for loading user programs so that larger user programs could be loaded and run more efficiently.

#### Documentation

The following documentation was produced under this contract: an FMS user guide and a DES user guide, prepared in the standard "lesson" format of all AUGMENT subsystem documentation; the FMS/DES System Programmer's Guide; and online Help information for FMS and DES.

Final Technical Report Summary:  
Conclusions Reached

CONCLUSIONS REACHED

Based on our work, it is clear to us that the notion of a coherent, consistent system for collecting and examining project financial and manpower data is viable. Additional work on the lowest level data storage and retrieval mechanisms (possibly making use of existing commercial systems suited to the needs of FMS/DES) would be useful. This is due to the fact that data storage and retrieval is based on the original design of the first experimental version and is inadequate for a widely used and highly interactive system as FMS has grown to be. Also, further development is necessary on the Template system in order to develop a more adequate (i.e., less complex) language for representing forms and the user interactions required to complete those forms. The other systems developed under this contract are useful in a variety of environments, but are especially well suited to the needs of the RADC project management process.

Final Technical Report Summary:  
Problems Encountered

PROBLEMS ENCOUNTERED

Problems in implementation have been due in part to continuing evolution of the AUGMENT system. For example, rearrangement of the system address space has made possible the creation of much larger specialized application programs. The size of the current system is in excess of 26K (without the associated Template system), whereas the user programs buffer in the previous version (NLS8.5) is only 22K. During the contract period, we have experienced changes of mainframe hardware and relocation of the ARC offices from SRI to Tymshare. Additionally, ARC-Tymshare has hired several new programmers in order to deal with personnel shortages. These programmers began working at ARC in September 1978 and were not able to contribute to new development work until late November; experienced programmers had to spend some time training them -- time not available for use on other projects. This paid off in the application of the new, larger staff on projects (including FMS) beginning in November.

While designing and programming, we discovered subtle complexities in the Template system and other tools under development. These experiences are documented and should be useful to those preparing similar systems.

Some difficulties were encountered due to changes in the personnel at RADC concerned with the use and development of FMS. Because of these changes, we received inadequate (and sometimes untimely) feedback on the design of the Manpower system and on drafts of user documentation. Response on the Variance Report task requirements was not forthcoming. Finally, computer service which was supposed to be GFE under this contract was not continuously delivered, leading to delays in the performance of the tasks outlined in the Statement of Work.

Final Technical Report Summary:  
General Methodology and Technical Results

**GENERAL METHODOLOGY**

All the subsystems were developed in AUGMENT using CML for specification of the user interfaces and L10 for coding of the execution functions. All existing AUGMENT tools are also available for use in this context.

**TECHNICAL RESULTS**

Technical results are the FMS, DES, and Manpower subsystems and supporting technical and user documentation. The source code for the developed systems resides on the Office-2 and Office-3 computers in the directory XFMS. Instructions for accessing this code will be provided upon request. The supporting documentation has been delivered.

Final Technical Report Summary:  
Implications for Further Research

IMPLICATIONS FOR FURTHER RESEARCH

Some possible further developments are primarily concerned with improving the efficiency of data handling. The data base management system upon which FMS/DES is based is built upon the AUGMENT structured file system. While suitable for a prototype system with small to medium-sized data bases, it is inadequate for a production system with large data handling requirements. Possible remedies for this include:

Modifying the data base itself to a compacted, more highly encoded form.

Use of other (commercially) existing data base systems that provide an efficient data storage and a process level interface.

The FMS/DES system will then provide the efficient user interface, with all its associated features (such as report generation, screen manipulation, etc.). Note that the data storage and retrieval portion need not reside on the same machine as the FMS system. Reaching through the ARPANET (or some commercial computer network) to existing commercial data base management systems is possible. The FMS system has been developed with this model in mind. Such reach-through capability has been demonstrated in other AUGMENT projects, but has not been used in a production data base management system environment.

Another important tool worth developing is the data base administrator tool. Currently, control of the system parameters, such as assignment of access control, data recovery in cases of malfunction, and other functions that relate to administering the use of the system, must be handled manually. The existence of such a tool will ensure that the smooth operation of the system, from the administrative standpoint, does not depend on a given individual who must be technically oriented.

Evaluation of Problem Areas:  
FMS/DES Subsystems

EVALUATION OF PROBLEM AREAS

FMS/DES SUBSYSTEMS

FMS is an online information system for maintaining, accessing, and analyzing a financial data base. It exists as two integrated subsystems under AUGMENT. The first subsystem, the Data Entry System (DES), is a data base tool that maintains precise control over the form and content of every data entry in a structured data base. DES use is limited to qualified data entry personnel and is unavailable to the general user; it is the only subsystem that can write information to the data base. The second subsystem is a user's tool, referred to as "FMS" because it is the most commonly used FMS subsystem. FMS allows direct user interaction, including retrieval and report generation, with the data maintained by FMS. It can only read and not change information in the data base.

The FMS data base is modeled on the RADC procurement process and has been designed to handle an "Effort" as its basic unit. An Effort is defined for FMS as the smallest unit of work at RADC. It usually has a one-to-one correspondence with a contract and is identified by a job order number; it may also represent in-house work. In FMS, Efforts are defined to be either "planned" or "actual". FMS can reflect the position of new Efforts within the procurement cycle.

An Effort consists of two levels in FMS. The uppermost level is referred to as the "Effort record" and contains data items that are unique to an Effort or a contract (e.g., contract title, project engineer, duration, contract face value, total manpower, contractor or performer, contract number). Subsumed beneath each Effort record are any number of Purchase Request (PR) records which are "owned" by the Effort record. The funding data that makes up a contract is found at the PR level. A PR record contains data about how money is allocated on a fiscal year basis (ICO data), the project and task supplying the money, the Program Element Code, the line number, and other information.

The previous versions of the FMS system were not based upon proper data base management system techniques. Rather they were developed to prove a point: that a data base management system can operate in symbiosis with other AUGMENT-based systems. As such the data base system was quite primitive. For example:

The data base was constructed much like a "personal" data base, i.e., where mostly one specific person (who understands the system in detail) uses the system most of the time, and the adaptability of the system to diverse users is not required.

## Evaluation of Problem Areas: FMS/DES Subsystems

The data in the data base files was stored as text. This is a straightforward but inefficient way to store data. It makes it harder to use data in its natural forms (integers, dates, etc.), requires more file space (resulting in data base compaction problems), and causes inefficiencies in the most basic data base operation -- the search.

Being a "personal" data base, it had very few safeguards to protect the integrity of the data. It was possible for users to access the same data concurrently for modification and sometimes accidentally override someone else's modification. If the system crashed the user had to go into the data base in NLS and edit the file by hand in order to make the data consistent again.

All the system's features were hard-coded. Thus, a change in the data structures or even the location of the files containing the system data required changing some code and recompiling the system. In such a data base management system it is preferable to have the names of all system files listed in a central location for easy modification. Moreover, there were few provisions for special protection on system files to insure system integrity.

Because it was a simple system, some of the algorithms used were not completely analyzed and no attention was given to the efficiency of these algorithms.

The activity on the current project can be classified into two main parts: restructuring the system and improving user capabilities. The following paragraphs elaborate on these.

### Restructuring FMS

The previous FMS system could be best characterized by ad hoc, inconsistent implementation. There was no distinct definition of the data base, the access mechanism, or the auxiliary mechanisms that process user commands. For example, adding a new field to the data base required modification of code in an inconsistent manner in different files. The necessity of being able to lock Efforts to protect against simultaneous update attempts by different users was recognized but inadequately solved; a user could inadvertently delete a locked Effort and thus have a record locked forever. The new approach taken this year is a giant step toward eliminating the inconsistencies that caused such problems.

### The Control Scheme

A new centralized control scheme has been implemented. A centralized control scheme is straightforward and particularly

Evaluation of Problem Areas:  
FMS/DES Subsystems

suited for nondistributed data bases. The control scheme we use is also good for a distributed data base where write operations are nondistributed, which is a likely configuration within RADC. Basic to the scheme is a central file that contains all the information the system needs to know about itself. Hence when the system is initialized, it accesses the central control file and retrieves all the relevant information. The system need only know the name and structure of the control file.

The following information is recorded in the control file:

Access information. The information necessary to enforce the access mechanism. (See The Access Mechanism below.)

Locking information. A basic data base management system operation is the locking of information; it prohibits two different users from simultaneously accessing the same record for modification. The centralized storage of the locks in the control file ensures that the user cannot modify the lock information (except through the system), and it can provide a picture of the activity to the data base administrator.

JON generator. The system must be able to provide, at the user's request, a unique job order number (JON) for an Effort (unique at least within the same TPO). The information necessary to generate such a number is stored in the control file.

File name mapping. In order to achieve complete separation between system features and those seen by the user, actual file names must be hidden from the user. The control file therefore contains the appropriate mapping between generic names as they appear to the user and actual file names, which are obviously system-dependent. This applies to all file names -- for TPOs, ledgers, and manpower files.

#### The Access Mechanism

The data contained in the FMS data base is considered sensitive data and therefore needs to be protected against intentional as well as inadvertent modifications. This problem is not unique to RADC, and a foolproof general solution does not exist; it is a research area. However, some measures to safeguard the data, although primitive, have been installed. The mechanism includes a standard password access procedure and a user access level assignment.

## Evaluation of Problem Areas: FMS/DES Subsystems

When entering the FMS or DES system, the user is required to provide a password before any access to the data is allowed. This password is distinct (and should, in fact, be different) from any other password the user is required to provide at any other stage of accessing the FMS system, such as the password used to log into the computer, or the AUGMENT identifier. This mechanism gives the data base administrator control over the set of people who have access to the data base.

Once the password is verified, the user is assigned a "level of access". This level of access is equivalent to assigning the user specific capabilities. The number of such levels is not limited; however, currently only two levels are used: read and read-write. It is envisioned that the level of access will be used to further classify users so that more sensitive commands (such as one which deletes Efforts) will be available to a more restricted subset of the users of the system.

### The Data Dictionary

In an attempt to separate the data base itself from the access tools, a data dictionary has been provided. This dictionary consists of a set of tables and system-provided procedures that supply information on the nature of all the appropriate data fields. The following paragraphs briefly describe the contents of the dictionary; a more complete description is provided in the FMS/DES System Programmer's Guide.

**Field attributes.** Each field in the data base has distinct attributes. The following are the supported attributes:

**Field name.** The official name by which the field is known to the system.

**Exist/notexist.** Since tables need not be compacted (i.e., can have holes), this is a means of finding whether a given token actually refers to a field.

**Physical/logical.** Physical fields are those that appear directly in the data base. Logical fields are those values that can be derived from other data in physical fields.

**Owning record.** An FMS field can belong to an Effort record, a PR record, or a manpower record. A field cannot belong to more than one record.

**Data type.** A field can be of type string (text), integer, floating point (decimal), dollar amount, or date.

Evaluation of Problem Areas:  
FMS/DES Subsystems

Shift case. Indicates whether the field contents must be all upper case.

Removable/nonremovable. Some fields may not be removed from the owning record.

Field syntax. Physical fields must adhere to a strict syntax to ensure that data is correctly stored and is meaningful when later retrieved. A mechanism has been installed to verify the syntax of any physical fields.

Field semantics. Every field in the data base has a very clear interpretation and therefore the set of values it may assume is limited. A mechanism has been installed to ensure the appropriate semantics of the field. Note, however, that there is only so much control that can be exerted on the user's input. For example, in numeric fields one can limit the values to a given range but one cannot, of course, ensure the validity of the data if it falls within that range. The semantic checking should be viewed more as an aid to the user rather than a data base protection mechanism.

Logical field extraction. Logical fields are those that are not explicitly stored in a record, but which can be evaluated based on other physical fields in this record or in related ones. An algorithm is therefore associated with each logical field that defines the way it is computed. Internally, these algorithms are represented by a set of "evaluation procedures", one for each logical field. Users do not have to access these algorithms directly since the low-level field retrieval procedures provide the appropriate mechanisms. Hence, the distinction (at retrieval time) between physical and logical fields is completely transparent.

#### Data Integrity

Typically, relations exist between various records and fields in the data base and take on different forms, and frequently these records and fields are stored in different data files. For example, moving an Effort from one TPO to another involves modifying both the original and new TPO files as well as all the ledger files that have any information pertaining to this Effort (even under several JONs). Or, the total time charged against an Effort that is recorded in the manpower data must be constrained by the total manpower allocation recorded in the Effort record (physically stored in a separate file). Ensuring such consistencies is regarded as data integrity.

## Evaluation of Problem Areas: FMS/DES Subsystems

Of course it is straightforward to assemble some of these constraints and enforce strict validation before any permanent changes are made to the data base. The problem becomes severe when uncontrolled yet likely events (such as file system error, machine crash, user abort by the attention key) occur in the midst of such an involved operation. Typically, one cannot predict the state of the data base in such cases. However, it is possible to minimize the time periods in which the data base is actually vulnerable to such mishaps.

The AUGMENT file system has been thoroughly studied, and to an extent augmented, to ensure that its best features are utilized. In addition, all the algorithms that modify the FMS data base have been modified to minimize access to the data base until all validations and calculations have occurred. The vulnerable periods have been minimized. A simple recovery mechanism (manual, so far) has been devised which can get the data base into a stable state with at most the last transaction lost in case of a system crash.

### Algorithm Improvements

The algorithms have been improved so that they are more efficient and also more general and expandable, thus allowing more capabilities. They are described in detail in the FMS/DES System Programmer's Guide.

### Improved User Capabilities

The items described in the System Restructuring section above are only indirectly visible to the end user. They manifest themselves in better response and more flexibility, and lend themselves to much easier administration tasks. The items described in this section are those that are more readily visible to the user. We do not specify here the details of all the user interface changes or command additions, but rather describe the capabilities that were added to the system. See the FMS and DES User Guides for further elaboration.

### Comprehensive Search Mechanism

A new, more general, search mechanism has been implemented. This mechanism allows the checking of any (independent) portions of the data base according to arbitrary criteria. The system recognizes the operators Not, Equal, Greater, Less, and any combination of these. In addition, the user can specify any operand for the search, including EMPTY (i.e., whether a given field is or is not specified).

Evaluation of Problem Areas:  
FMS/DES Subsystems

A complementary sorting mechanism has been implemented that allows the reordering of the data identified in the search according to a user-specified criterion.

Ledger Tracing

The FMS system keeps a ledger of all the transactions made against the data base. This data is recorded in separate data files called the "ledger files". As part of this year's activity we developed a ledger tracing algorithm that identifies, in order, all ledger records that relate to a given contract (even under JON changing circumstances). This capability is demonstrated by the History command (as described in the FMS User Guide). The algorithm used is very general and lends itself to preparation of special ledger-based summaries such as variance reports, and to certification algorithms that can help ensure data base integrity.

Multiple File Capability

A mechanism has been implemented to mix, under various circumstances, data that comes from different sources. In the current system, this mechanism is used in the simulation process (see Simulation below), and in the ability to view, in a mixed fashion, regular Effort and PR data along with related manpower data which is stored separately. The generality of the algorithm lends itself to efficient retrieval of data from various sources (different data bases on the same machine, different data bases accessible cross-network) and eventually to support of a fully distributed data base.

Archival

The nature of the FMS data base is that the amount of data is always increasing, leading to extremely large files, slowing down of most algorithms, and retention of superfluous data. However, due to the nature of the data (i.e., the fact that it is generally composed of time-limited contractual information), much data becomes irrelevant when a contract terminates and all transactions against it have been completed (usually within a year of contract termination). Hence, data that is no longer relevant could be separated from the data base. A mechanism has been designed and implemented to assemble all data relevant to a contract, to remove these from the data base, and to collectively archive them. (Relevant data includes the current Effort record, all related PRs, and all ledger transactions including cross-JON changes.)

Evaluation of Problem Areas:  
FMS/DES Subsystems

Simulation

It is often important for the data base user to temporarily modify portions of the data base so that projections or analysis can be made. This is called "simulation". A simulation mechanism has been implemented in the system.

The FMS simulation mechanism allows the user to identify any independent portions of the data base (by identifying each record separately, by performing searches, etc.) as the subject for simulation. The user can then arbitrarily modify any and all fields of the records on a temporary basis. Appropriate commands, permitting summation over fields and report generation, can then create reports based on the actual or the simulated data. In fact, the user can perform the same operation on both data and then compare the results.

A complementary sorting mechanism has been implemented that allows the reordering of the simulated data according to a user-specified criterion.

The algorithms that implement the simulation mechanism are general enough so that future applications requiring simulation can readily use the existing mechanism.

Evaluation of Problem Areas:  
Manpower Subsystem

**MANPOWER SUBSYSTEM**

A mechanism has been implemented to record and retrieve manpower data related to other data stored in the FMS data base. All FMS retrieval mechanisms are cognizant of this new data base, and fields from it can be retrieved and mixed with regular data. Additional fields have been added to the Effort record to report summaries of the manpower data that relate to that Effort. The Template mechanism (see below) is also cognizant of these. A complementary tool, the FORM2 subsystem, has been designed and implemented to facilitate the collection and entry of manpower data.

The FORM2 subsystem is designed to record, modify, transfer, and view employee work time in specified Efforts during a month's period. After the end of a monthly work period, the total time spent by an individual against a particular Effort is calculated and entered in the FMS data base. Each employee is restricted to one time card, i.e., a time card must be completed and updated, and the data integrated into FMS, before another one can be created and accessed.

Forms are accessed by user identifiers and should only be used with the FORM2 subsystem. Because of the size of the display screen the user can view only nine days of data at a time. The user may specify at any time the starting date for the nine days of data to be seen.

In the FORM2 subsystem, commands exist to: access a form2 for a specified user identifier, month, and year; enter information into the form; delete information from the form or delete the entire form; and copy, move, or update the form. The form2 format and the subsystem commands are described in Appendix II.

Evaluation of Problem Areas:  
Template/Forms System

TEMPLATE/FORMS SYSTEM

ARC has continued work on and modified a general purpose form generation system. Part of this system is a template writing language for specifying locations of information on pages or terminals and styles of data collection from users or data management systems. It also permits the establishment of contingencies among fields in one or more forms.

The Fill subsystem is built upon primitives in the AUGMENT Template system. It provides an interface and protocol for filling out instances of forms whose templates were written in the Template language. The AUGMENT version of the FMS system makes use of the Fill subsystem to generate reports; it includes a "Generate" command which has been implemented to show the feasibility of automatically generating a report from the FMS data base.

The template primitives are arranged into three principal groupings:

A scanner and interpreter use coroutine linkages to parse and interpret information contained in templates written in a Template definition language. Specifications permit creation of data elements from information obtained from users after suitable promptings or from items contained in other files or data bases. These specifications define the portrayal of the information and subsequent storage of information in data bases.

A portrayal module takes information obtained and generated by the interpreter concerning the current value of a field and its location in a form, retrieval and storage information concerning the field in one or more data bases, and other information, and places it in special properties within the AUGMENT file structure. In essence the file created with these properties becomes a compiled version of the template: a specific instance of the form.

The principal difficulty with the current experimental Template system is the complexity of the template definition language. To be really useful there will have to be special subsystems to aid in the creation of the more simple templates. Additionally, programming and debugging aids for what is truly a powerful programming language are needed. There is also a need to complete implementation of some planned but until now unimplemented features of the template language.

A modified sample RADC 74 (JB form) template was created to show the feasibility of generating reports automatically from the FMS data base. Special features of the Template system were used to create data elements that obtained information from the FMS data base.

Evaluation of Problem Areas:  
Template/Forms System

Appendix III contains the current syntax of the template description language as well as the sample template for RADC 74.

Evaluation of Problem Areas:  
Modifications to AUGMENT

MODIFICATIONS TO AUGMENT

In order to implement the systems developed under this contract, some minor modifications had to be made to AUGMENT. One of these was the modification of address spaces so that larger user programs could be loaded and run more efficiently. The mechanism by which user programs are loaded was modified. Additionally, a design for an Overlay system was undertaken, but was not implemented under the FMS contract due to resource limitations.

Evaluation of Problem Areas:  
Documentation

DOCUMENTATION

The following documentation was produced under this contract:

An FMS User Guide and a DES User Guide, prepared in the standard "lesson" format of all AUGMENT subsystem documentation.

The FMS/DES System Programmer's Guide.

Online Help information for FMS and DES.

Printed copies of the FMS and DES User Guides, phototypeset using the AUGMENT Output Processor, and the System Programmer's Guide have been delivered. The Help information may be accessed through the Help command on Office-2 or Office-3; it is kept in the files named FMS and DES in the USERGUIDES directory.

Appendix I:  
Summary of New Commands and Features in FMS/DES

**APPENDIX I  
SUMMARY OF NEW COMMANDS AND FEATURES IN FMS/DES**

The following lists new commands and features implemented in FMS/DES and the associated data base as part of this contract. (The documentation provided under this contract gives more detail.)

**New Fields**

Effort records: MP-TOTAL, MP-YTD, MP-PREVYEARS, PREVIOUS-JON (ledger only), REASON (ledger only)

PR records: PR-TITLE, PRMANHOURS

Manpower records: IDENT, YTD, PREV-YEARS, PERIOD

**New Features**

Simulation, sorting, ledger tracking, archiving, searching

**New Commands**

Copy  
Delete (Effort)  
Finish  
Generate  
History  
Increment/Decrement  
Remove (field)  
Simulate  
Sort

**Improved Commands**

Find

Appendix II:  
The Manpower Subsystem Form2 Format and Commands

**APPENDIX II  
THE MANPOWER SUBSYSTEM FORM2 FORMAT AND COMMANDS**

The format of the form2 file created by the Manpower (FORM2) subsystem is as follows:

1) General form information

a) File identifier

**EMPLOYEE TIME EXPENDITURE**

b) Update information

(1) The hours per Effort for each Effort in the form have been calculated. (N implies no; Y implies yes.)

(2) The hours per day for each day in the month have been calculated. (N implies no; Y implies yes.)

(3) The hours per Effort for each Effort in the form have been entered into FMS. (N implies no; S implies started; Y implies yes.)

c) Employee and time period information

(1) Name (last, first)

(2) Month and year

d) Column headings

(1) Effort column

(2) One column for each day of the month

(3) Total hours per Effort column

(4) Related TPO for the Effort

2) Effort entries

a) Effort identifier

b) For each day of the month, the number of hours spent in the Effort

Appendix II:  
The Manpower Subsystem Form2 Format and Commands

- c) Total hours spent in the Effort for the month (done at update time)
  - d) Related TPO for the Effort
- 3) Total hours worked per day (done at update time)

**Entering and Leaving the FORM2 Subsystem**

The FORM2 subsystem is entered as an AUGMENT subsystem using the Goto command. When entered, the subsystem will attempt to load the calculator routines and create a new display area. If the attempt is successful, the message "FORM2 subsystem is now available. Access a form2 to continue." is presented in the status window and the user may continue. If the attempt is not successful, the message "FORM2 subsystem cannot be used." is presented and the subsystem is exited.

When exited, the subsystem will return the user to the user's status prior to entering the subsystem.

**Commands and Results**

The subsystem commands are Access, Start-printing-at-day, Enter, Delete, Transfer, and Update.

**Access command**

This command accesses a form2 for a specified user identifier, month, and year. The user must also specify the first day for which data will be displayed on the screen. The data for the next eight days of the month will also be displayed.

If the specified identifier is a valid identifier:

- 1) If no form2 exists, one is created for the specified month and year.
- 2) If a form2 exists and it is for the specified month and year, it is accessed and displayed.
- 3) If a form2 exists and it is NOT for the specified month and year, a message is printed. The form is not accessed.

If the specified identifier is not valid, a message is printed and another identifier is requested.

Appendix II:  
The Manpower Subsystem Form2 Format and Commands

**Start-printing-at-day command (A form2 must be accessed)**

This command formats the display of the form2 such that the day specified is the first day for which data will be displayed on the screen. The data for the next eight days of the month will also be displayed.

**Enter command (A form2 must be accessed)**

For a specified day, Effort, and TPO, this command enters a specified hour time in the form2 currently accessed.

**Delete command (A form2 must be accessed)**

The Delete Effort command deletes from the form2 currently accessed the Effort record specified by Effort identifier and related TPO.

The Delete Form command deletes the current form2 file and all its versions.

Note that the Delete command is to be used with extreme care. Once something is deleted it cannot be recovered.

**Transfer command (A form2 must be accessed)**

This command transfers the form currently accessed to a specified address. The transfer may be a Copy or a Move. The Copy will simply copy the form to the specified location. The Move will copy the form to the specified location and delete the form in the form2 file.

The transferred data will always be AUGMENT statements; therefore, the data should always be transferred to an AUGMENT file. No filtering of data or line truncation according to current view-specs is done.

**Update command (A form2 must be accessed)**

If the form has already been updated, this command prints a message and terminates.

If the form has not been updated:

- a) If the total hours per Effort for each Effort in the form have NOT been calculated, this command calculates the total

Appendix II:  
The Manpower Subsystem Form2 Format and Commands

hours per Effort for each Effort in the form, enters the totals in the form, and sets the appropriate update status in the form.

b) If the total hours per day for each day of the month have NOT been calculated, this command calculates the total hours per day for each day of the month, enters the total hours per day statement in the form, and sets the appropriate update status in the form.

c) If all the Effort records have NOT been updated into FMS, this command enters into FMS the total hours per Effort for each Effort in the form, not already entered into FMS, and sets the appropriate update status in the form. Entries into FMS are done by user identifier, TENEX date of last day in month, hour value in floating point, Effort identifier by string address, and related TPO by string address.

#### Limitations and Future Plans

It would be desirable to have more than one form2 per user.

#### Changes to User's Directory

File creation. A file is created when a form is accessed for an ident and none exists. The form is created for the month and year specified by the user.

File modifications. Modifications are done by the Enter, Delete, and Update commands. Whenever a file is closed, it is updated to a new version.

File structure. The form2 file has the structure described below.

An origin statement:

```
< directory name, ident-FORM2.NLS;version number, >, day-month-
year hour:min (creation date and time) ident ;;;
```

A header statement:

```
EMPLOYEE TIME EXPENDITURE  EFF TOT = N; DAY TOT = N; FMS UPDATE
= N;
```

Last name, First name month year

```
JON      1      2...days...TOTALS: RELATED TPO
```

Appendix II:  
The Manpower Subsystem Form2 Format and Commands

**Effort statements:**

EFFORT...(hour times).....(total):effort tpo

Day total statement (only after update):

##TOTALS...(hour times)...

The following depicts the format of the form2 file:

< directory name, IDENT-FORM2.NLS;2, >, 2-Feb-79 15:43 IDENT ;;;;

EMPLOYEE TIME EXPENDITURE EFF TOT = Y; DAY TOT = Y; FMS UPDATE = Y;

Last name, First name FEB 1976

JON	1	2	3	4	5	6	7	8	9	10	
11	12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	TOTALS:RELATED TPO				

\*99954000

000.00 :5

\*99955000

000.00 :5

\*99956000

000.00 :5

##TOTALS 00.00 00.00 00.00 00.00 00.00 00.00 00.00 00.00 00.00 00.00  
00.00 00.00 00.00 00.00 00.00 00.00 00.00 00.00 00.00 00.00  
00.00 00.00 00.00 00.00 00.00 00.00 00.00 00.00 00.00 00.00

Appendix III:  
Template Language Syntax and Sample RADC 74 Template

APPENDIX III  
TEMPLATE LANGUAGE SYNTAX AND SAMPLE RADC 74 TEMPLATE

Template Language Syntax (Draft)

The following syntax is subject to revision as the system is still evolving.

```
template := '( templatename ') TEMPLATE [defaultrule] enddelim 1$spec
END.
defaultrule := [< errorrule, emptyrule >] [loadrule] $compoindrue
spec := noisetext / leftdelim specrule rightdelim / leftdelim
specrulelink rightdelim
noisetext := character-string
leftdelim := '!
rightdelim := '! / '@ / '#
enddelim := ';
loadrule := LOAD link $(', link ) enddelim
specrule := [<emptyrule, errorrule>] compoindrue
compoindrue := 1$simplerule
simplerule := <whilerule, caserule, formatrule, resultrule, assignment,
blockrule, abortrule, attributerule, callrule, performrule,
nullrule>
whilerule := WHILE boolexp DO compoindrue enddelim
caserule := CASE expression OF 1$(boolean-relation expression ':'
compoindrue enddelim) ENDCASE compoindrue enddelim
attributerule := <updaterule / mandatoryrule / namerule>
callrule := CALL procname '([expression $(), expression]) '
performrule := PERFORM specrulelink
nullrule := NULL
updaterule := UPDATE / NOUPDATE
mandatoryrule := MANDATORY / NOMANDATORY
namerule := FIELDNAME string-expression
datarule := retrieverule / entryrule / userrule / procedurerule
blockrule := BLOCK <heightrule, widtrule, blockstartrule, cpmod,
left, right, top, bottom> enddelim / LINE <widtrule, cpmod, left,
right, top, bottom> enddelim
formatrule := <justification, hiliterule, formatcontrol>
hiliterule := HILITE / NOHILITE
assignment := variable '_ expression enddelim
emptyrule := EMPTY compoindrue enddelim
errorrule := ERROR compoindrue enddelim
abortrule := ABORT abortmessage
userrule := SELECT selector noiseword % prompt %
procedurerule :=(INTEGER/STRING/BOOLEAN) PROCEDURE procname '(
[expression $(), expression]) '
blockstartrule := START arithmetic-expression
resultrule := (USE/EXCEPTION) expression enddelim
```

Appendix III:  
Template Language Syntax and Sample RADC 74 Template

```
heightrule ::= HEIGHT (height (FIXED/DYNAMIC) / INFINITY)
widthrule ::= WIDTH (width, (FIXED/DYNAMIC) / INFINITY)
cpmod ::= POSITION [ [&]x-exp ][, [&]y-exp ]
x-exp ::= arithmetic-expression
y-exp ::= arithmetic-expression
justification ::= JUSTIFICATION (LEFT/RIGHT/CENTER/...)
left ::= LEFT arithmetic-expression
right ::= RIGHT arithmetic-expression
top ::= TOP arithmetic-expression
bottom ::= BOTTOM arithmetic-expression
height ::= arithmetic-expression
width ::= arithmetic-expression
noiseword ::= string-expression
prompt ::= string-expression
expression ::= arithmetic-expression / string-expression
abortmessage ::= string-expression
variables ::= V 1$2D
arithmetic-expression ::= arfactor [ arop arfactor ]
arfactor ::= variable / datarule / arconstant / arinternals
arop ::= '+ / '-' / '*' / '/'
arconst ::= number
arinternals ::= XPOSITION / YPOSITION / WIDTH / HEIGHT / TOP / BOTTOM
/ LEFT / RIGHT
string-expression ::= strfactor ['& strfactor ]
strfactor ::= variable / datarule / string-constant
string-constant ::= '"' characterstring '"'
boolexp ::= expression [ booleanrelation expression ]
booleanrelation ::= '=' / '#' / '<' / '>' / "<=" / ">="
selector ::= TEXT / CHARACTER / WORD / VISIBLE / INVISIBLE / BRANCH /
STATEMENT / PLEX / GROUP / NUMBER
```

To be defined:

```
retrieverule
entryrule
formatcontrol
```

Appendix III:  
Template Language Syntax and Sample RADC 74 Template

Sample RADC 74 (JB Form) Template

Note: The sample template is not definitive; it does not necessarily represent the optimal use of the template language. However, it gives the flavor of some of the complexity and possibilities of the Template system, which provides the opportunity for specifying with some exactness the style of user interaction in filling out a form.

```
(F74a) TEMPLATE
DEFAULT !jb-default!
    STATUS REPORT  !orig! !update-nr  ! AS OF      JOCAS NR. MASIS
ACC.NR.
                                         !as-of  ! !jon  !
!masis-acc-nr!
TITLE                               CENTER PROGRAM MANAGER
SYMBOL/EXT
!title                           !!center-program-manager!
!symbol-ext !
PE:          DIRECTIVE:
TPO/THRUST
!pe        !!directive
!!tpo!/!thrust!
ULTIMATE CUSTOMER(S)/USER(S):      LAST MASIS
UPDATE:
!ultimate-customer-user
update!
DESCRIPTION
!description!
STATUS/PROGRESS
!status-progress!
APP/SC CHG!chg!DATE PE      TYPE SOURCE PRI FY FY-1 CFY FF FY+1 TO
COMP TOTAL
FCSTMP
CHG!mpchg!!chdate!!pe1!!ty!!src!!pfy!!lfy!!cfy!!ff!!infy!!cmp!!ttl!
LITERATURE SEARCH
!pe2!!ty!!src!!pfy!!lfy!!cfy!!ff!!infy!!cmp!!ttl!
MASIS!masis!DDC!ddc!
!pe2!!ty!!src!!pfy!!lfy!!cfy!!ff!!infy!!cmp!!ttl!
STINFO!stinfo! OTHER!other!      FORECAST MANPOWER
!fpr!!fis!!fcr!   X  !fnx!!ftc!!ftt!
DATE:!!sdate!  ACTUAL MANPOWER  !apr!!als!!acr!   X   X   X
!att!
REMARKS !remarks!                  YR  !yr
!
MO
!m1!!m2!!m3!!m4!!m5!!m6!!m7!!m8!!m9!!m10!!m11!!m12!!m13!!m14!!m15!
!m16!
PREWARD ASSMNT PR NR !pr-nr  !!cs10!
```

Appendix III:  
Template Language Syntax and Sample RADC 74 Template

```
STP F'CAST LSR CSR  INIT DATE!init-date!!cs9!
!st1!!fcast!!lsr!!csr! CONTRACTOR
!st2!! fcast2  !! !IX !!contractor    !!cs8!

!!!      !!X !!  !!          !!cs7!
           CONTRACTOR NR
!!!      !!X !!  !!contractor-nr !!cs6!

16!fcast5!!  !!  !!          !!cs5!
RECYCLE !recycle!DATE !rdate  !TYPE !type  !!cs4!
POST AWARD ASSESSMENT COST
!post-award-assess!!cost      !!cs3!
TECH PERF      START DATE
!tech-perf!!  !! !start-date  !!cs2!
FINANCIAL      COMPL DATE
!financial!!  !! !compl-date  !!cs1!
SCHEUDGE      PROGRAM SCHEDULE
!schedule !!  !! !program-schedu  !
MANNING
!manning !!  !!  !!          !
LOGISTICS
!logistics!!  !!  !!          !
TESTING
!testing !!  !!  !!          !
KEY DECISIONS
key-decis!!  !!  !!          !
PROG DIRECTION
!prog-dire!!  !!  !!          !
DOCUMENTATION
!documenta!!  !!  !!          !

OVERALL ASSESS
!overall-as!!  !!  !!          !
CPM % TECH COMP
!cpm-%-tech-comp  !!          !
CONTR % TECH COMP
!contr-%-tech-comp  !!          !
           REVIEW AND APPROVAL      FINAL DD FORM 250 SIGNED
DATE
!review-and-approval      !!final-dd-form-250-
si!!date  !
           LSR CSR ACTION DIRECTED AT LSR ACTION DIRECTED  OPR
(SYMBOL)ACTION COM
!  !!lsr !!csr !!action-directed-at-1!!action-direct  !!opr-symbol
!!action-
LEV          SIGNATURE/TITLE/SYMBOL    SUSPENSE DATE
PLETED DATE
```

Appendix III:  
Template Language Syntax and Sample RADC 74 Template

```
!le!! !! !!signature-title-symb!!suspense-date
!!pleted-
    DATE DATE                               SPECIFY ACTION IF REQUIRED
    ! !date!!date!!                           !!specify-action-if-required
    !!
    !           DATE
    !           !date
    !!
END.
```

F74-ITEMS

```
JB-DEFAULT
    LOAD <xsubsys,dmydate.tmplt,>;
    UPDATE
    BLOCK
        WIDTH 20 FIXED
        HEIGHT 1 FIXED;
    EMPTY
        CASE V3 OF
            > 0: V15 _ RETRIEVE rendseq(V3);;
            ENDCASE NULL;
            ;
            EXCEPTION "none";
            USE "empty";
            ;
    FCAST5
        USE RETRIEVE rgetfield("OBLDATE", V13);
    TPO
        BLOCK WIDTH 2 FIXED;
        BLOCK POSITION 60, 6;
        V11 _ 5;
        USE V11;
    V11 _ STRING PROCEDURE rgettponam(V1);;
CASE V1 OF =0:
    V16 _ SELECT ANSWER "This TPO Number is undefined. Use DES to
    enter a new one. Do you wish to reenter this one?";
    CASE V16 OF
        = 1: V1 _ SELECT TEXT "TPO number";
        ENDCASE ABORT "aborted";
        ;
    ;
ENDCASE ;
JON
EMPTY
    EXCEPTION "none";
    USE "empty";
    V3 _ 0;
```

Appendix III:  
Template Language Syntax and Sample RADC 74 Template

```
;  
BLOCK WIDTH 12 FIXED;  
V12 _ SELECT TEXT "Eight character Job Order Number";  
V2 _ RETRIEVE rnextrinsequence("", V1);  
V13 _ RETRIEVE rgeteffort(V12, V2);  
V3 _ RETRIEVE rgtsub("", V13);  
USE V12;  
V3 _ RETRIEVE rgetprseq("", V13);  
EMPTY EXCEPTION  
V16 _ SELECT ANSWER "This Job Order Number is undefined. Use  
DES to enter a new one. Do you wish to reenter this one?";  
CASE V16 OF  
    = 1: PERFORM !jon!;  
    ENDCASE ABORT "aborted";  
    ;  
    V15 _ RETRIEVE rendseq(V3);  
    ;  
ORIG  
    V10 _ SELECT CHARACTER "Is this the original Status Report?";  
CASE V10 OF  
    = 1: USE "x";;  
    ENDCASE USE " ";;  
BLOCK WIDTH 6 FIXED;  
USE " ORIG";  
UPDATE-NR  
BLOCK WIDTH 12 FIXED;  
USE "UPDATE # ";  
CASE V10 OF  
    = 0: USE SELECT WORD "Update Number";;  
    ENDCASE USE " ";;  
AS-OF  
EMPTY EXCEPTION "none";;  
BLOCK WIDTH 10 FIXED;  
USE SELECT TEXT "Date of review (Example: 11-AUG-84)";  
USE SELECT TEXT "Date of review  
(Example: 11-AUG-84 / today / yesterday / tomorrow)";  
THRUST  
BLOCK WIDTH 3 FIXED;  
V14 _ STRING PROCEDURE rgetthrust(V13);  
USE V14;  
V14 _ FLOAT PROCEDURE rgetthrust(V13);  
EMPTY EXCEPTION SELECT TEXT "THRUST (Example: 8.5)";;  
MASIS-ACC-NR  
EMPTY EXCEPTION "none";;  
BLOCK WIDTH 13 FIXED;  
USE RETRIEVE rgetfield("ACCESSION", V13);  
TITLE  
EMPTY EXCEPTION "none";;
```

Appendix III:  
Template Language Syntax and Sample RADC 74 Template

```
BLOCK WIDTH 40 FIXED;
USE RETRIEVE rgetfield("TITLE", V13);
CENTER-PROGRAM-MANAGER
EMPTY EXCEPTION "none";;
BLOCK WIDTH 22 FIXED;
USE RETRIEVE rgetfield("ENGINEER", V13);
SYMBOL-EXT
EMPTY EXCEPTION "none";;
BLOCK WIDTH 13 FIXED;
USE SELECT TEXT "Center Program Manager's SYMBOL/EXT
Example: ISIM/3857";
PE
EMPTY
EXCEPTION "none";
USE "empty";
V3 = 0;
;
BLOCK WIDTH 14 FIXED;
V18 = RETRIEVE rgetfield("PEC", V3);
USE V18;
V17 = RETRIEVE rnextrinsequence("", V3);
V18 = RETRIEVE rgetfield("PEC", V17);
V15 = RETRIEVE rendseq(V3);
DIRECTIVE
EMPTY EXCEPTION "none";;
BLOCK WIDTH 45 FIXED;
USE SELECT TEXT "DIRECTIVE
Example: IS Memo dated 8-Aug-77";
ULTIMATE-CUSTOMER-USER
EMPTY EXCEPTION "none";;
BLOCK WIDTH 51 FIXED;
USE SELECT TEXT "Ultimate Customers and Users
Example: AFSC, SAC, ARMY.";
LAST-MASIS-UPDATE
EMPTY EXCEPTION "none";;
BLOCK WIDTH 24 FIXED;
USE SELECT TEXT "Last Masis Update
Example: 11-Aug-77";
DESCRIPTION
EMPTY EXCEPTION "none";;
BLOCK
WIDTH 75 FIXED
HEIGHT 5 FIXED;
EXCEPTION "none";
USE RETRIEVE rgetfield("EFFORT-WRITEUP", V13);
USE "Waiting for effort-writeup file.";
STATUS-PROGRESS
EMPTY EXCEPTION "none";;
```

Appendix III:  
Template Language Syntax and Sample RADC 74 Template

```
BLOCK
    WIDTH 75 FIXED
    HEIGHT 5 FIXED;
    USE SELECT TEXT "Status / Progress
Limit about 50 words?";

CHG
    BLOCK WIDTH 3 FIXED;
    V15 _ SELECT CHARACTER "APP/SCOPE change?";
    CASE V15 OF
        = 1: USE " x";;
        ENDCASE USE "   ";
MPCHG
    BLOCK WIDTH 3 FIXED;
    CASE V15 OF
        = 1: USE "   ";
        ENDCASE
            V16 _ SELECT CHARACTER "Forecast MP change?";
            CASE V16 OF
                = 1: USE "x";;
                ENDCASE USE "   ";
;
CHDATE
    BLOCK WIDTH 5 FIXED;
    CASE V16 OF
        = 1: USE SELECT TEXT "Date of change";;
        ENDCASE
            CASE V15 OF
                = 1: USE SELECT TEXT "Date of change";;
                ENDCASE USE "   ";
;
PE1
    EMPTY EXCEPTION "none";;
    BLOCK WIDTH 6 FIXED;
    USE V18;
PE2
    EMPTY
        EXCEPTION "none";
        USE "empty";
        V3 _ 0;
;
    BLOCK WIDTH 9 FIXED;
    CASE V3 OF
        = 0: USE " --";;
        ENDCASE
            V17 _ RETRIEVE rgtsuc("", V3);
            USE RETRIEVE rgetfield("PEC", V17);
;
EMPTY
```

Appendix III:  
Template Language Syntax and Sample RADC 74 Template

```
V15 _ RETRIEVE rendseq(V3);
;
CASE V3 OF
= 0: USE "--";
ENDCASE
V17 _ RETRIEVE rnextinsequence("", V3);
USE RETRIEVE rgetfield("PEC", V17);
;
TY
EMPTY EXCEPTION "none";;
BLOCK WIDTH 6 FIXED;
CASE V3 OF
= 0: USE "--"
ENDCASE
USE RETRIEVE rgetfield("TY", V17);;
SRC
EMPTY EXCEPTION "none";;
BLOCK WIDTH 5 FIXED;
CASE V3 OF
= 0: USE "--"
ENDCASE
USE RETRIEVE rgetfield("SOURCE", V17);;
PFY
EMPTY EXCEPTION "none";;
BLOCK WIDTH 6 FIXED;
CASE V3 OF
= 0: USE "--"
ENDCASE
USE RETRIEVE rgetfield("PFY", V17);;
LFY
EMPTY EXCEPTION "none";;
BLOCK WIDTH 6 FIXED;
CASE V3 OF
= 0: USE "--"
ENDCASE
USE RETRIEVE rgetfield("LFY", V17);;
CFY
EMPTY EXCEPTION "none";;
BLOCK WIDTH 6 FIXED;
CASE V3 OF
= 0: USE "--"
ENDCASE
USE RETRIEVE rgetfield("CFY", V17);;
FF
EMPTY EXCEPTION "none";;
BLOCK WIDTH 6 FIXED;
CASE V3 OF
= 0: USE "--"
```

Appendix III:  
Template Language Syntax and Sample RADC 74 Template

```
ENDCASE
    USE RETRIEVE rgetfield("FF", V17);;
NFY
    EMPTY EXCEPTION "none";;
    BLOCK WIDTH 6 FIXED;
    CASE V3 OF
        = 0: USE " --"
    ENDCASE
        USE RETRIEVE rgetfield("NFY", V17);;
CMP
    EMPTY EXCEPTION "none";;
    BLOCK WIDTH 6 FIXED;
    CASE V3 OF
        = 0: USE " --"
    ENDCASE
        USE RETRIEVE rgetfield("CMP", V17);;
TTL
    EMPTY EXCEPTION "none";;
    BLOCK WIDTH 6 FIXED;
    CASE V3 OF
        = 0: USE " --"
    ENDCASE
        USE RETRIEVE rgetfield("TOTAL", V17);;
MASIS
    BLOCK WIDTH 1 FIXED;
    V15 _ SELECT CHARACTER "LITERATURE SEARCH: check MASIS?";
    CASE V15 OF
        = 1: USE "x";;
    ENDCASE USE " ";;
DDC
    BLOCK WIDTH 1 FIXED;
    V15 _ SELECT CHARACTER "LITERATURE SEARCH: check DDC?";
    CASE V15 OF
        = 1: USE "x";;
    ENDCASE USE " ";;
STINFO
    BLOCK WIDTH 1 FIXED;
    V15 _ SELECT CHARACTER "LITERATURE SEARCH: check STINFO?";
    CASE V15 OF
        = 1: USE "x";;
    ENDCASE USE " ";;
OTHER
    BLOCK WIDTH 1 FIXED;
    V15 _ SELECT CHARACTER "LITERATURE SEARCH: check OTHER?";
    CASE V15 OF
        = 1: USE "x";;
    ENDCASE USE " ";;
```

Appendix III:  
Template Language Syntax and Sample RADC 74 Template

```
LSDATE BLOCK WIDTH 16 FIXED; USE SELECT TEXT "LITERATURE SEARCH:  
type DATE";  
FCR  
    BLOCK WIDTH 6 FIXED;  
    USE SELECT TEXT "FORECAST MANPOWER: Current fiscal year";  
FLS  
    BLOCK WIDTH 6 FIXED;  
    USE SELECT TEXT "FORECAST MANPOWER: Last fiscal year";  
FPR  
    BLOCK WIDTH 6 FIXED;  
    USE SELECT TEXT "FORECAST MANPOWER: Prior fiscal years";  
FNX  
    BLOCK WIDTH 6 FIXED;  
    USE SELECT TEXT "FORECAST MANPOWER: Next fiscal years";  
FTC  
    BLOCK WIDTH 6 FIXED;  
    USE SELECT TEXT "FORECAST MANPOWER: To Complete";  
FTT  
    BLOCK WIDTH 6 FIXED;  
    USE SELECT TEXT "FORECAST MANPOWER: Total";  
ACR  
    BLOCK WIDTH 6 FIXED;  
    USE SELECT TEXT "ACTUAL MANPOWER: Current fiscal year";  
ALS  
    BLOCK WIDTH 6 FIXED;  
    USE SELECT TEXT "ACTUAL MANPOWER: Last fiscal year";  
APR  
    BLOCK WIDTH 6 FIXED;  
    USE SELECT TEXT "ACTUAL MANPOWER: Prior fiscal years";  
ATT  
    BLOCK WIDTH 6 FIXED;  
    USE SELECT TEXT "ACTUAL MANPOWER: Total";  
REMARKS  
    BLOCK  
        WIDTH 35 FIXED;  
        LENGTH 7 FIXED  
        USE SELECT TEXT "REMARKS";  
COST SCHEDULE ANALYSIS BLOCK WIDTH 39 FIXED; USE SELECT TEXT "COST  
SCHEDULE ANALYSIS";  
YR BLOCK WIDTH 5 FIXED; USE SELECT TEXT "YR";  
MO BLOCK WIDTH 5 FIXED; USE SELECT TEXT "MO";  
COST-SCHEDULE-ANALYSIS BLOCK WIDTH 5 FIXED; USE SELECT TEXT  
"COST/SCHEDULE ANALYSIS";  
PREAWARD-ASSESSMENT BLOCK WIDTH 19 FIXED; USE SELECT TEXT  
"PREAWARD ASSESSMENT";  
PR-NR BLOCK WIDTH 16 FIXED; USE RETRIEVE rgetfield("PR-NUMBER",  
V13);  
ST1 BLOCK WIDTH 2 FIXED; USE SELECT TEXT "STP";
```

Appendix III:  
Template Language Syntax and Sample RADC 74 Template

FORECAST BLOCK WIDTH 8 FIXED; USE SELECT TEXT "FORECAST";  
LSR BLOCK WIDTH 4 FIXED; USE SELECT TEXT "LSR";  
CSR BLOCK WIDTH 4 FIXED; USE SELECT TEXT "CSR";  
INIT-DATE BLOCK WIDTH 16 FIXED; USE SELECT TEXT "INIT DATE:";  
CONTRACTOR BLOCK WIDTH 16 FIXED; USE SELECT TEXT "CONTRACTOR";  
CONTRACTOR-NR BLOCK WIDTH 16 FIXED; USE SELECT TEXT "CONTRACTOR  
NR";  
16 BLOCK WIDTH 2 FIXED; USE SELECT TEXT "16";  
RECYCLE BLOCK WIDTH 7 FIXED; USE SELECT TEXT "RECYCLE";  
DATE BLOCK WIDTH 12 FIXED; USE SELECT TEXT "DATE:";  
TYPE BLOCK WIDTH 16 FIXED; USE SELECT TEXT "TYPE:";  
POST-AWARD-ASSESSMENT BLOCK WIDTH 19 FIXED; USE SELECT TEXT "POST  
AWARD ASSESSMENT";  
COST BLOCK WIDTH 16 FIXED; USE SELECT TEXT "COST:";  
TECH-PERF BLOCK WIDTH 11 FIXED; USE SELECT TEXT "TECH PERF";  
START-DATE BLOCK WIDTH 16 FIXED; USE SELECT TEXT "START DATE:";  
FINANCIAL BLOCK WIDTH 11 FIXED; USE SELECT TEXT "FINANCIAL";  
COMPL-DATE BLOCK WIDTH 16 FIXED; USE SELECT TEXT "COMPL DATE:";  
SCHEDULE BLOCK WIDTH 11 FIXED; USE SELECT TEXT "SCHEDULE";  
PROGRAM-SCHEDULE BLOCK WIDTH 16 FIXED; USE SELECT TEXT "PROGRAM  
SCHEDULE";  
MANNING BLOCK WIDTH 12 FIXED; USE SELECT TEXT "MANNING";  
LOGISTICS BLOCK WIDTH 12 FIXED; USE SELECT TEXT "LOGISTICS";  
TESTING BLOCK WIDTH 12 FIXED; USE SELECT TEXT "TESTING";  
KEY-DECISIONS BLOCK WIDTH 12 FIXED; USE SELECT TEXT "KEY  
DECISIONS";  
PROG-DIRECTION BLOCK WIDTH 12 FIXED; USE SELECT TEXT "PROG DIREC-  
TION";  
DOCUMENTATION BLOCK WIDTH 12 FIXED; USE SELECT TEXT "DOCUMENTA-  
TION";  
OVERALL-ASSESS BLOCK WIDTH 12 FIXED; USE SELECT TEXT "OVERALL  
ASSESS";  
CPM-%-TECH-COMP BLOCK WIDTH 20 FIXED; USE SELECT TEXT "CPM % TECH  
COMP:";  
CONTR-%-TECH-COMP BLOCK WIDTH 20 FIXED; USE SELECT TEXT "CONTR %  
TECH COMP:";  
REVIEW-AND-APPROVAL BLOCK WIDTH 42 FIXED; USE SELECT TEXT "REVIEW  
AND APPROVAL";  
FINAL-DD-FORM-250-SIGNED BLOCK WIDTH 22 FIXED; USE SELECT TEXT  
"FINAL DD FORM 250 SIGNED";  
DATE: BLOCK WIDTH 13 FIXED; USE SELECT TEXT "DATE:";  
LSR BLOCK WIDTH 6 FIXED; USE SELECT TEXT "LSR";  
CSR BLOCK WIDTH 6 FIXED; USE SELECT TEXT "CSR";  
ACTION-DIRECTED-AT-LSR BLOCK WIDTH 22 FIXED; USE SELECT TEXT "AC-  
TION DIRECTED AT LSR";  
ACTION-DIRECTED BLOCK WIDTH 15 FIXED; USE SELECT TEXT "ACTION DI-  
RECTED";  
OPR SYMBOL BLOCK WIDTH 15 FIXED; USE SELECT TEXT OPR (SYMBOL)";

Appendix III:  
Template Language Syntax and Sample RADC 74 Template

ACTION COMPLETED DATE BLOCK WIDTH 9 FIXED; USE SELECT TEXT "ACTION  
COMPLETED DATE";  
LEV BLOCK WIDTH 4 FIXED; USE SELECT TEXT "LEV";  
SIGNATURE-TITLE-SYMBOL BLOCK WIDTH 22 FIXED; USE SELECT TEXT  
"SIGNATURE/TITLE/SYMBOL";  
SUSPENSE-DATE BLOCK WIDTH 30 FIXED; USE SELECT TEXT "SUSPENSE  
DATE:";  
DATE BLOCK WIDTH 6 FIXED; USE SELECT TEXT "DATE";  
DATE BLOCK WIDTH 6 FIXED; USE SELECT TEXT "DATE";  
DATE BLOCK WIDTH 20 FIXED; USE SELECT TEXT "DATE";  
SPECIFY-ACTION-IF-REQUIRED BLOCK WIDTH 31 FIXED; USE SELECT TEXT  
"SPECIFY ACTION IF REQUIRED (Continue on reverse side if needed)";

F74-VARIABLE-KEY

V1: TPO stid (initialized)  
V2: stid of first Effort in sequence  
V3: handle of first PR  
V10: original flag  
V11: TPO number (name)  
V12: 8 char Job Order Number (JON)  
V13: our Effort stid  
V14: Thrust  
V15: temporary  
V16: temporary  
V17: current PR stid  
V18: first PEC  
V19: PR exist flag

**MISSION**  
**of**  
**Rome Air Development Center**

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C<sup>3</sup>I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

